REMARKS

This Amendment is submitted in response to the Office Action of August 5, 2008 (hereinafter "the Office Action"). Upon entry of this Amendment, claims 2-4, 6-19, 21-33, 35-44, 46, and 47 remain pending, and new claims 48 and 49 are added.

All references to the claims, except as noted, will be made with reference to the claim list above beginning on page 2. All references to "the Office Action," except as noted, will be referencing the most recent Office Action dated August 5, 2008. If there is any confusion or questions regarding any aspect of this Amendment, the Examiner is invited to contact the undersigned.

Amendment

Independent claims 2, 6, and 35 are amended to more explicitly specify that the verification of an instruction is performed dynamically, i.e., during execution of a series of instructions including the specific instruction being verified. Claim 7 is amended to improve clarity. Amendments are made to certain additional dependent claims for consistency. New claims 48 and 49 are added, and generally correspond to existing claims 38 and 39, but depend from claim 2, except that claim 48 explicitly specifies that the hash value is a function of all contents of the memory block. No new matter is introduced by this Amendment.

Claim Rejections - 35 U.S.C. §112

Claim 7 stands rejected under 35 U.S.C. §112, second paragraph for being indefinite. Specifically, the Office Action specifies that "The term, 'potentially' is considered indefinite by examiner" (Office Action, page 2, lines 14-16). Applicants respectfully traverse. Applicants disagree with the contention in the Office Action that "potentially" is per se indefinite. Nevertheless, Applicants have amended claim 7 to replace the offending language with "indeterminate portion." This phrase is intended to convey variability of the portion of the memory block being masked.

Claim Rejections - 35 U.S.C. § 103(a)

Claims 2-4, 6-19, 21-28, 30-32, and 35-44 stand rejected under 35 U.S.C. § 103(a) for being obvious over U.S. Patent 5,826,013 issued to Nachenberg ("N1") in view of U.S. Patent 6,021,510 issued to Nachenberg ("N2"). Claims 29, 33, 46, and 47 stand rejected under 35

U.S.C. § 103(a) as being obvious over N1 in view of N2 and the article entitled "Complete Computer System Simulation: The SimOS Approach" by Mendel Rosenblum et al. (hereinafter, "SimOS"). Applicants respectfully traverse because the prior art references would not have suggested to a person having ordinary skill in the art at the time the invention was made each and every feature set forth in the claims.

The primary reference N1 relates to virus scanning software capable of detecting polymorphic viruses. The software emulates execution of instructions within a file so that, if a virus is present, the virus' own decryption routine can be run prior to scanning the file to expose a static virus body of a polymorphic virus. Once the static virus body is exposed it can be scanned to determine if the file is or contains a known virus.

The secondary reference N2 relates to software for maintaining a list of data blocks on a disk that are known to be free of viruses, so that files formed by the disk blocks can be identified as clean without re-scanning unless the block is written to or the virus signature database is updated.

Neither of the references teach or suggest dynamic software verification. As set forth in the specification, "the invention provides . . . for verifying the validity of code dynamically by comparing the contents of a unit (or sub-unit) of a memory (such as a page) that contains code ***currently being executed.***" (paragraph 30) (emphasis added). The references do not show performing verification during execution of the software; or deciding as to whether to allow execution based on the signature. Applicants have amended the claims to more explicitly incorporate "dynamically" into the body of the claims. Claim 2 now requires that the steps of "determining . . . an identifying value for a memory block that contains the next instruction" and "determining . . . whether the identifying value satisfies a validation condition" occur "during the executing of the series of instructions." Independent claims 6 and 35 are similarly (though not identically) so modified to explicitly provide that validity verification occurs dynamically, i.e., during execution of the program being verified.

For obviousness under 35 U.S.C. § 103(a), each and every limitation must be suggested by the prior art reference, or references when combined or modified (MPEP 2143). Since the prior art references do not teach dynamically verifying validity of executing code, Applicants respectfully submit that the independent claims are not obvious, and depending claims are not obvious for at least the same reasons as the independent claims.

The dependent claims further distinguish the invention from the prior art. For example, with regard to claims 4 and 37, the Office Action cites N1, element 474 (presumably as represented in Figure 4B), stating, "viral signature match determines if the hash value of the instruction is different than all of the reference values" (Office Action, page 4, lines 11-14). However, the viral signature referenced in N1 is not the signature or hash value of a memory block, but are instead virus signatures, which are simply "code segments unique to different viruses" (see N1, col. 5, lines 43-46).

Claims 6, 39, and 40 (and new claim 49) include an operation involving ignoring (masking) part of a memory block when computing a hash of the memory block. The Office Action cites N1, col. 12, lines 20-30, stating "the selected page locations are scanned and the non-selected ones are ignored" (Office Action, page 5, lines 9-14). Applicants respectfully disagree. Specifically, N1 does not teach generating a hash code from tagged locations of a memory block as suggested in the Office Action. Instead, tagged locations of a ***file*** are scanned to see if ***code segments*** (not ***hash values***) match signatures in the virus database. N1 does not specify that the tagged memory locations be in a common memory block, nor that a ***hash value*** of a memory block is ***ever calculated***. Virus scanning, as described in N1 involves directly comparing code to a signature that is merely a code segment, not a hash value of a code segment. Thus, the "signature" described in N1 is nothing more than a code segment, and N1 therefore does not suggest hashing a portion of a page.

Claim 7 sets forth, "identifying potentially non-constant contents of the current memory block, the non-constant contents being valid but changeable so that they do not indicate validity of the current memory block as a whole" and "configuring the mask so that the mask designates at least the non-constant contents to be ignored when generating the hash value. . . ." The Office Action cites Figure 5 of N1 and specifically element 512 thereof, stating "checks for register modifications" and "exclude all viruses that cannot perform memory write with non-initialized index register" (Office Action, page 6, lines 3-9). Applicants respectfully disagree. N1 describes tracking modifications to memory locations caused by the emulation of virus instructions that decrypt the body of a virus. Each location in memory that is written to therefore potentially contains viral code. Claim 7 specifies that the mask designates that the "non-constant contents . . . be ignored when generating the hash value." N1 in contrast specifically checks these locations in memory. The excluding of viruses that cannot perform memory write described in the Office Action does not refer to excluding locations in memory from scanning, but excluding viruses from the database that

the tagged memory locations can potentially match. I.e., virus scanning involves comparing scanned code with known virus signatures to exclude viruses as being possible matches. If all the signatures are excluded, then the scanned code is considered virus free. See e.g., the discussion of static and dynamic exclusion modules 230, 240 in the paragraph bridging columns 6 and 7 in N1.

The above discussion of dependent claims is not exhaustive and merely presents examples of differences between Applicants' claimed invention and cited references. Applicants respectfully request independent reconsideration of the outstanding rejections of the independent claims and each of the dependent claims.

Applicants respectfully submit that the present Application is now in condition for allowance. A Notice of Allowance is therefore respectfully requested.

If the Examiner has any questions concerning the present amendment, the Examiner is kindly requested to contact the undersigned.

Respectfully submitted,
VMware, Inc.

/Leonard Heyman/

Leonard Heyman
Reg. No. 40, 418

Telephone: 650-427-2390